



Computational analysis of particle-laden-airflow erosion and experimental verification

Alessio Castorrini^{1,6} · Paolo Venturini² · Alessandro Corsini² · Franco Rispoli² · Kenji Takizawa³ · Tayfun E. Tezduyar^{4,5}

Received: 27 November 2019 / Accepted: 16 February 2020 / Published online: 24 March 2020
© The Author(s) 2020

Abstract

Computational analysis of particle-laden-airflow erosion can help engineers have a better understanding of the erosion process, maintenance and protection of turbomachinery components. We present an integrated method for this class of computational analysis. The main components of the method are the residual-based Variational Multiscale (VMS) method, a finite element particle-cloud tracking (PCT) method with ellipsoidal clouds, an erosion model based on two time scales, and the Solid-Extension Mesh Moving Technique (SEMMT). The turbulent-flow nature of the analysis is addressed with the VMS, the particle-cloud trajectories are calculated based on the time-averaged computed flow field and closure models defined for the turbulent dispersion of particles, and one-way dependence is assumed between the flow and particle dynamics. Because the target-geometry update due to the erosion has a very long time scale compared to the fluid–particle dynamics, the update takes place in a sequence of “evolution steps” representing the impact of the erosion. A scale-up factor, calculated based on the update threshold criterion, relates the erosions and particle counts in the evolution steps to those in the PCT computation. As the target geometry evolves, the mesh is updated with the SEMMT. We present a computation designed to match the sand-erosion experiment we conducted with an aluminum-alloy target. We show that, despite the problem complexities and model assumptions involved, we have a reasonably good agreement between the computed and experimental data.

Keywords Particle-laden flow · Erosion · Residual-based VMS · Particle-cloud tracking · Ellipsoidal cloud

1 Introduction

Turbomachinery blades are quite often subjected to particle-laden-flow erosion. If the blade does not have sufficient surface protection, the erosion can damage it to the point of altering its aerodynamics, degrading the performance of turbomachinery system. Computational analysis of blade erosion is becoming inevitable in reducing the prototyping

cost. Reliable computational analysis can help the designers find new blade protection strategies. It can also help them to plan effective maintenance schedules. Earlier, closely-related studies [1,2] focused on the prediction of rain erosion patterns for a full-span wind-turbine blade. In [3,4], the focus was on the effect of the erosion on the aerodynamic performance of the blade, highlighting the correlation between the erosion patterns and geometry evolution.

✉ Alessio Castorrini
alessio.castorrini@unibas.it

Paolo Venturini
paolo.venturini@uniroma1.it

Alessandro Corsini
alessandro.corsini@uniroma1.it

Franco Rispoli
franco.rispoli@uniroma1.it

Kenji Takizawa
Kenji.Takizawa@tafsm.org

Tayfun E. Tezduyar
tezduyar@tafsm.org

¹ Scuola di Ingegneria, Università della Basilicata, Viale dell’Ateneo Lucano 10, 85100 Potenza, Italy

² Dipartimento di Ingegneria Meccanica e Aerospaziale, Sapienza University of Rome, Via Eudossiana 18, 00184 Rome, Italy

³ Department of Modern Mechanical Engineering, Waseda University, 3-4-1 Ookubo, Shinjuku-ku, Tokyo 169-8555, Japan

⁴ Mechanical Engineering, Rice University – MS 321, 6100 Main Street, Houston, TX 77005, USA

⁵ Faculty of Science and Engineering, Waseda University, 3-4-1 Ookubo, Shinjuku-ku, Tokyo 169-8555, Japan

The typical approach we see in the literature does not account for the changes in the flow field due to the geometry change. In most studies the flow computation is performed only on the original geometry, and then the computed flow field is used for predicting the particle transport and the particle erosion/deposit on the blade surface. It was shown in [5] that different blade geometries, at the same operational point and for the same power output, lead to different local aerodynamic fields with very different erosion patterns. Therefore we expect that even a minor modification on the critical parts of a blade section can make a noticeable difference in the flow field. Furthermore, the aerodynamics affects the particle motion and thus, the erosion itself. To account for this effect and to increase the fidelity of the blade erosion computational analysis, an integrated method was developed in [3] for predicting the time evolution of the interaction between the fluid–particle dynamics and blade erosion and geometry change.

This class of applications are characterized by small particle concentration. Therefore one-way dependence is assumed between the flow and particle dynamics, that is, particle (and cloud) motion is driven by the flow but the flow is not influenced by the particles. The concept of one-way dependence has been used in other computational engineering analyses. For example, in [6], the concept is used for computing the aerodynamic forces acting on the suspension lines of spacecraft parachutes, where the suspension lines are assumed to have no influence on the flow. In [7], the same assumption is used to study the particle–shock interaction. In [8–10], the assumption is used in flow-driven string dynamics in turbomachinery, where the strings are assumed to have no influence on the flow.

The main components of the integrated method given in [3] are the Streamline-Upwind/Petrov-Galerkin (SUPG) [11] and Pressure-Stabilizing/Petrov-Galerkin (PSPG) [12] stabilizations, a finite element particle-cloud tracking (PCT) method [1–3] with one-way dependence, an erosion model based on two time scales, and the solid-extension mesh moving technique (SEMMT) [13–17].

The combination of the SUPG and PSPG stabilizations is called “SUPS” [17]. In [3], the turbulent-flow nature of the analysis was addressed with a Reynolds-Averaged Navier–Stokes (RANS) model, with the Navier–Stokes equations discretized based on the SUPS stabilization and the two RANS closure equations discretized based on the SUPG stabilization. Here, we address the turbulent-flow nature of the analysis with the residual-based variational multiscale (VMS) method [17–21]. The VMS has two additional sta-

bilization terms beyond those in the SUPS and therefore subsumes the SUPS. It has good turbulence modeling features. We compute the unsteady aerodynamic field until we reach a fully-developed flow, and then calculate from that the time-averaged flow variables. Methods like the SUPS and VMS involve stabilization parameters. There are various ways of calculating these parameters (see, for example, [2,3,7,22–43]). We will describe in a later section what stabilization parameters we use here.

The PCT method was first formulated by Baxter [44], and then further developed [1,2,45–47] and improved to obtain statistically-independent results [48]. The trajectory of the particle-cloud center is calculated with finite element discretization. Of the elements of that “particle mesh,” we use the ones inside the cloud, which has a trajectory-dependent size. The tracking method accounts for the drift-velocity gradient in the near-wall regions [46,49]. In [3], the clouds were assumed to be of spherical shape. Here, we generalize the cloud shape to an ellipsoid with independent semi-axis lengths. The particle-cloud trajectories are calculated based on the time-averaged flow field and a closure model for the turbulent dispersion of particles. The closure model in [3] was based on the turbulence closure variables. Here, we propose a closure model based on the diagonal components of the Reynolds stress tensor.

Tabakoff and coworkers [50–52] were among the earliest researchers working on numerical prediction of erosion in turbomachinery. Their methods and experiments served as a foundation for other studies in the field. Nowadays several empirical and semi-empirical erosion models can be found in the literature. The model used in [2] for rain erosion was a modified version of the Keegan model [53]. The models used in [3] were defined in terms of the mass eroded per unit mass of the impacting particles. For the rain erosion, the model from Springer et al. [54] was used, and for the sand erosion, the model from Oka et al. [55]. Here, we use the same sand erosion model. We determine the empirical parameters associated with the target material by curve fitting to the data from the experiment.

The geometry update due to the erosion has a very long time scale compared to the fluid–particle dynamics [4,56,56]. Therefore a single time-marching computation with the typical time-step size of the flow computation is not practical. Instead, we use a sequence of “evolution steps” to represent the impact of the erosion. We time-discretize the geometry evolution not in terms of a standard time step, but in terms of a threshold erosion value that we expect to alter the blade aerodynamics from its current operation pattern or a threshold operation period that we expect to be long enough to alter the blade aerodynamics.

The computation associated with an evolution step gives us the erosion distribution for a specific particle size, for a specific number of particles injected to the computational

⁶ Department of Engineering, Lancaster University, Gillow Avenue, Lancaster LA1 4YW, UK

domain. We have two approaches for scaling-up the erosion distribution at each evolution step. A. We scale-up the erosion distribution by a factor that raises its maximum value to the threshold erosion value. We use the same factor to scale-up the number of particles to the actual number of particles for that evolution step. B. We scale-up the number of particles by a factor that raises it to the actual number of particles for the threshold operation period. We use the same factor to scale-up the erosion distribution.

As the target geometry evolves, we update the mesh with the SEMMT. The core mesh moving method in the SEMMT is the Jacobian-based stiffening method [57–60]. In the core method, the motion of the internal nodes is determined by solving the equations of linear elasticity. The mesh deformation is dealt with selectively based on the sizes of the elements. The selective treatment is based on the way we account for the Jacobian of the transformation from the element domain to the physical domain. The objective is to stiffen the smaller elements, which are typically placed near solid surfaces, more than the larger ones. When the method was introduced in [57–59], it consisted of simply dropping the Jacobian from the finite element formulation of the mesh moving (elasticity) equations. This results in the smaller elements being stiffened more than the larger ones. In the SEMMT, the thin layers of elements placed near solid surfaces are treated almost like an extension of the solid structure. In solving the equations of elasticity governing the motion of the fluid nodes, higher stiffness is assigned to the thin layers of elements compared to the other fluid elements. Two ways of accomplishing this were proposed in [13]: solving the elasticity equations for the nodes connected to the thin layers of elements separately from the elasticity equations for the other nodes, or together.

To show how the integrated method works, we perform a computation designed to match the sand-erosion experiment we conducted with an aluminum-alloy target.

In Sect. 2, we provide an overview of the integrated method. Section 3 is an overview of the mathematical model, including the PCT model. The residual-based VMS is given in Sect. 4. In Sect. 5, we describe the discretized particle equations, including the turbulent dispersion of particles. The erosion models and erosion thickness computation, including how the scale-up factors are calculated, are described in Sect. 6. The experiments are described in Sect. 7. The computation is presented in Sect. 8, and the concluding remarks are given in Sect. 9.

2 An integrated method

We presented in [3] an integrated method to simulate the long-term erosion of a wind-turbine blade. The method is made of a multiphase flow solver coupled with a geometry

update method. We use here much of the same method, altering only some of the sub-steps. Therefore, some parts of this section, included for completeness, are from [3].

The time scales associated with the unsteady aerodynamics and turbulence, particle transport and dynamics, erosion of the target material and the change in the geometry that produces a significant variation in the average flow field and particle trajectories are different. The geometry update due to the erosion has a very long time scale compared to the fluid–particle dynamics, making a single time-marching simulation with the typical time-step size of the flow computation not practical. The basic idea is to have a sequence of evolution steps to represent the impact of the erosion. We time-discretize the evolution of the geometry not in terms of a standard time step, but in terms of a threshold erosion value that we expect to alter the target-geometry aerodynamics from its current operation pattern or a threshold operation period that we expect to be long enough to alter the target-geometry aerodynamics. The alteration of the flow patterns leads to the alteration of the particle dynamics, which in turn alters the erosion patterns.

An evolution step is composed of the sub-steps listed below in the order they are taken.

1. Compute the flow field with the residual-based VMS.
2. Compute the particle-cloud trajectories with the PCT method [3], updated as presented in Sect. 5.
3. Compute the erosion distribution over the target surface.
4. Scale-up the computed erosion distribution using the threshold erosion value or operation period that triggers a geometry alteration.
5. Update the geometry and fluid mechanics mesh to the next configuration. The mesh update is done with the SEMMT.

The computation associated with an evolution step gives us the distribution of the erosion thickness e for a specific particle size, for a specific number of particles injected to the computational domain. Depending on the application, we have two approaches for scaling-up e at each evolution step.

- A. We scale-up e by a factor that raises its maximum value e_{max} to the threshold erosion value e_{thr} . We use the same factor to scale-up the number of particles to the actual number of particles for that evolution step. At the end of all the evolution steps, we obtain a correlation map from the damaged configurations to the amount of particles needed to produce those configurations. This map can later be used to estimate by interpolation the geometrical configuration resulting from the amount of particles in a specific application.

B. We scale-up the number of particles by a factor that raises it to the actual number of particles for the threshold operation period. We use the same factor to scale-up e . At the end of all the evolution steps, we obtain a correlation map from the actual number of particles to the damaged configurations. This map can later be used to directly estimate the damaged configurations from a specific rate of exposure to particles during a long, specific period (e.g., 25 years) that we want to know the damage for.

3 Mathematical model

3.1 Navier–Stokes equations of incompressible flows

Let $\Omega \subset \mathbb{R}^{nsd}$ be the spatial domain with boundary Γ , and $(0, T)$ be the time domain. The Navier–Stokes equations of incompressible flows can be written on Ω and $\forall t \in (0, T)$ as

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) - \nabla \cdot \boldsymbol{\sigma} = \mathbf{0}, \quad (1)$$

$$\nabla \cdot \mathbf{u} = 0, \quad (2)$$

where ρ and \mathbf{u} are the density and velocity. The stress tensor $\boldsymbol{\sigma} = -p\mathbf{I} + 2\mu\boldsymbol{\varepsilon}(\mathbf{u})$, where p is the pressure, \mathbf{I} the unit tensor, $\mu = \rho\nu$ is the viscosity, ν the kinematic viscosity, and the strain rate $\boldsymbol{\varepsilon}(\mathbf{u}) = (\nabla \mathbf{u} + \nabla \mathbf{u}^T)/2$. The essential and natural boundary conditions associated with Eq. (1) are represented as $\mathbf{u} = \mathbf{g}$ on Γ_g and $\mathbf{n} \cdot \boldsymbol{\sigma} = \mathbf{h}$ on Γ_h , where Γ_g and Γ_h are the subsets of the boundary Γ , \mathbf{n} is the unit outward normal vector, and \mathbf{g} and \mathbf{h} are given functions.

3.2 Dispersed-phase model

Some parts of this section, included for completeness, are from [3]. Particle trajectories are simulated in a Lagrangian reference frame. Since particle concentration in this class of applications is very small (less than 10^{-6} in the particle volume fraction), a one-way dependence approach can be used [61]. We use the PCT model [44] to simulate a large number of particles without tracking them individually. The PCT approach was used in turbulent particle dispersion [45, 48, 62–64] and validated in turbomachinery and biomass furnaces [65, 66]. In the PCT model, each trajectory is not for a particle, but for a group (“cloud”) of particles, thus represents the evolution of the cloud position as a function of time:

$$\mathbf{x}_c = \int_0^t \mathbf{v}_c dt' + (\mathbf{x}_c)_0. \quad (3)$$

Here, subscript c refers to the cloud, \mathbf{v}_c is the velocity of the cloud, and $(\mathbf{x}_c)_0$ is the initial position of the cloud, which is the inflow boundary in our computations.

The equation of motion for the cloud is given by the Basset–Boussinesq–Oseen formulation, which, with one-way dependence hypothesis according to Armenio and Fiorotto [67], reads as

$$\frac{d\mathbf{v}_c}{dt} = \tau_R^{-1} (\langle \mathbf{u} \rangle - \mathbf{v}_c) + \left(1 - \frac{\rho}{\rho_p} \right) \mathbf{a}_{\text{GRAV}}, \quad (4)$$

where $\langle \cdot \rangle$ denotes ensemble average (defined later), ρ_p is the particle material density, \mathbf{a}_{GRAV} is the gravitational acceleration, and τ_R is the particle relaxation time, which, for spherical particles, is

$$\tau_R^{-1} = \frac{3}{4d_p} C_D \frac{\rho}{\rho_p} \|\langle \mathbf{u} \rangle - \mathbf{v}_c\|. \quad (5)$$

Here, d_p is the particle diameter and C_D is the drag coefficient based on the particle Reynolds number $Re_p = \frac{\|\langle \mathbf{u} \rangle - \mathbf{v}_c\| d_p}{\nu}$, first introduced in [68]. The Stokes number is defined as

$$Stk = \frac{\tau_R U}{L}, \quad (6)$$

where U is the free-stream velocity and L is the flow length scale. The initial value of \mathbf{v}_c is given as $\mathbf{v}_c(0) = \langle \mathbf{u} \rangle|_{t=0}$.

The ensemble average for the dispersed phase within the cloud is defined according to the hypothesis of independent statistical events, and for any ensemble-averaged quantity θ it reads as

$$\langle \theta \rangle = \frac{\int_{\Omega_c} \theta PDF(\mathbf{x}, t) d\Omega}{\int_{\Omega_c} PDF(\mathbf{x}, t) d\Omega}. \quad (7)$$

Here, Ω_c is the cloud domain and $PDF(\mathbf{x}, t)$ is the probability density function.

In the PCT approach, the particle position distribution within a cloud is assumed to be Gaussian, and the cloud size varies in time, depending on the flow behaviour. In [3], the cloud was assumed to be of spherical shape. The PDF of the particle distribution within the cloud was given¹ as

$$PDF(\mathbf{x}, t) = \frac{1}{\sqrt{(2\pi)^{nsd} \sigma^{nsd}}} e^{-\frac{1}{2} \left(\frac{\|\mathbf{x} - \mathbf{x}_c\|}{\sigma} \right)^2}, \quad (8)$$

where σ is the square root of the variance of particle position, and the cloud radius is 3σ .

Here, we generalize the cloud shape to an ellipsoid with nsd independent semi-axis lengths. Then the particle-distribution PDF becomes

$$PDF(\mathbf{x}, t) = \frac{1}{\sqrt{(2\pi)^{nsd} \det \mathbf{S}}} e^{-\frac{1}{2} (\mathbf{x} - \mathbf{x}_c) \mathbf{S}^{-1} (\mathbf{x} - \mathbf{x}_c)}, \quad (9)$$

¹ In the expression actually given in [3], the superscript nsd was missing. This inconsequential typo was also in [1, 2, 5, 46, 47].

where S is the covariance matrix of particle position. We will define it in Sect. 5. The semi-axis lengths of the ellipsoid are $3\sigma_1, 3\sigma_2$ and $3\sigma_3$, where σ_1^2, σ_2^2 and σ_3^2 are the eigenvalues of S , from the largest to the smallest. The eigenvectors are the corresponding axis directions. Because the semi-axis lengths are independent, the ellipsoid shape can change in time.

Combining Eqs. (4) and (5), we obtain

$$\frac{d\mathbf{v}_c}{dt} = C'_D \|\mathbf{u}\| - \mathbf{v}_c \cdot (\mathbf{u} - \mathbf{v}_c) + \left(1 - \frac{\rho}{\rho_p}\right) \mathbf{a}_{\text{GRAV}}, \tag{10}$$

where

$$C'_D = \frac{3}{4d_p} C_D \frac{\rho}{\rho_p}. \tag{11}$$

4 Residual-based VMS

4.1 Formulation

The residual-based VMS formulation [17] of Eqs. (1) and (2) is given as

$$\begin{aligned} & \int_{\Omega} \mathbf{w}^h \cdot \rho \left(\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \nabla \mathbf{u}^h \right) d\Omega + \int_{\Omega} \boldsymbol{\varepsilon}(\mathbf{w}^h) : \boldsymbol{\sigma} d\Omega \\ & - \int_{\Gamma_h} \mathbf{w}^h \cdot \mathbf{h}^h d\Gamma + \int_{\Omega} q^h \nabla \cdot \mathbf{u}^h d\Omega \\ & + \sum_{e=1}^{n_{el}} \int_{\Omega^e} \frac{\tau_{SUPS}}{\rho} \left(\rho \mathbf{u}^h \cdot \nabla \mathbf{w}^h + \nabla q^h \right) \cdot \mathbf{r}_M d\Omega \\ & + \sum_{e=1}^{n_{el}} \int_{\Omega^e} \nu_{LSIC} \nabla \cdot \mathbf{w}^h \rho r_C d\Omega \\ & - \sum_{e=1}^{n_{el}} \int_{\Omega^e} \tau_{SUPS} \mathbf{w}^h \cdot \left(\mathbf{r}_M \cdot \nabla \mathbf{u}^h \right) d\Omega \\ & - \sum_{e=1}^{n_{el}} \int_{\Omega^e} \frac{\tau_{SUPS}^2}{\rho} \mathbf{r}_M \cdot \left(\nabla \mathbf{w}^h \right) \cdot \mathbf{r}_M d\Omega = 0, \end{aligned} \tag{12}$$

where

$$\mathbf{r}_M = \rho \left(\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \nabla \mathbf{u}^h \right) - \nabla \cdot \boldsymbol{\sigma}^h, \tag{13}$$

$$r_C = \nabla \cdot \mathbf{u}^h \tag{14}$$

are the residuals of the momentum equation and incompressibility constraint, and τ_{SUPS} and ν_{LSIC} are the stabilization parameters (see Sect. 4.2). The test functions associated with \mathbf{u} and p are \mathbf{w} and q . A superscript “ h ” indicates that the function is coming from a finite-dimensional space. The superscript “ e ” is the element counter, and n_{el} is the number of elements.

4.2 Stabilization parameters

This section, included for completeness, is mostly from [3]. We first define the element length [69] in the advection-dominated limit:

$$h_{UGN} = 2 \left(\sum_{a=1}^{n_{en}} |\mathbf{s} \cdot \nabla N_a| \right)^{-1}, \tag{15}$$

where $\mathbf{s} = \frac{\mathbf{u}}{\|\mathbf{u}\|}$, n_{en} is the number of element nodes, and N_a is the basis function associated with node a . In the diffusion-dominated limit, the element length [23] is defined as

$$h_{RGN} = 2 \left(\sum_{a=1}^{n_{en}} |\mathbf{r} \cdot \nabla N_a| \right)^{-1}, \tag{16}$$

where \mathbf{r} is the unit vector in the direction of the solution gradient:

$$\mathbf{r} = \frac{\nabla \|\mathbf{u}\|}{\|\nabla \|\mathbf{u}\|\|}. \tag{17}$$

The components of τ_{SUPS} corresponding to the advection-, transient-, and diffusion-dominated limits were defined in [23,70] as

$$\tau_{SUGN1} = \left(\sum_{a=1}^{n_{en}} |\mathbf{u} \cdot \nabla N_a| \right)^{-1} = \frac{h_{UGN}}{2\|\mathbf{u}\|}, \tag{18}$$

$$\tau_{SUGN2} = \frac{\Delta t}{2}, \tag{19}$$

$$\tau_{SUGN3} = \frac{h_{RGN}^2}{4\nu}. \tag{20}$$

From these, τ_{SUPS} is defined as

$$\tau_{SUPS} = \left(\frac{1}{\tau_{SUGN1}^2} + \frac{1}{\tau_{SUGN2}^2} + \frac{1}{\tau_{SUGN3}^2} \right)^{-\frac{1}{2}}, \tag{21}$$

and ν_{LSIC} is defined [23] as

$$\nu_{LSIC} = \tau_{SUPS} \|\mathbf{u}^h\|^2. \tag{22}$$

Some new options for the stabilization parameters used with the SUPS and VMS were proposed in [24,30,31,34,36,39,42,71]. These include a fourth τ component, “ τ_{SUGN4} ” [36], which was introduced for the VMS, considering one of the two extra stabilization terms the VMS has compared to the SUPS. They also include stabilization parameters [36] for the thermal-transport part of the VMS for the coupled incompressible-flow and thermal-transport aligns. The element lengths and stabilization parameters introduced

in [39,71] target isogeometric discretization but are also applicable to finite element discretization. The stabilization parameters given in this subsection have node-numbering invariance for all element types, but they do not extend to isogeometric discretization in an ideal way. The element length expression introduced in [42] has node-numbering invariance for all element types, including simplex elements.

5 Discretized particle equations

The early parts of this section, included for completeness, are from [3]. In the discretized particle equations, ensemble averaging is carried out over the discretized cloud domain $\Omega_c = \bigcup_{e=1}^{n_{elc}} \Omega_c^e$, where Ω_c^e is the cloud element and n_{elc} is the number of elements. The cloud elements come from a fixed mesh, which we call “particle mesh,” and consist of the elements of the fixed mesh within a radius of 3σ . With that, the discretized version of ensemble averaging is written as

$$\langle \theta \rangle^h = \frac{\sum_{e=1}^{n_{elc}} \int_{\Omega_c^e} \theta PDF(\mathbf{x}, t) d\Omega}{\sum_{e=1}^{n_{elc}} \int_{\Omega_c^e} PDF(\mathbf{x}, t) d\Omega}, \tag{23}$$

where the element-level integration is performed by Gaussian quadrature.

5.1 Trajectory calculation

Spatially-discretized version of Eq. (10) is written as

$$\frac{d\mathbf{v}_c^h}{dt} = \mathbf{a}_c^h, \tag{24}$$

where

$$\mathbf{a}_c^h = C'_D \|\langle \mathbf{u} \rangle^h - \mathbf{v}_c^h\| \left(\langle \mathbf{u} \rangle^h - \mathbf{v}_c^h \right) + \left(1 - \frac{\rho}{\rho_p} \right) \mathbf{a}_{GRAV}. \tag{25}$$

Time discretization of Eq. (24) is performed with a predictor–multicorrector algorithm.

Predictor stage:

$$(\mathbf{v}_c^h)^0_{n+1} = (\mathbf{v}_c^h)_n + (\mathbf{a}_c^h)_n \Delta t. \tag{26}$$

Multicorrector stage:

$$(\mathbf{v}_c^h)^{i+1}_{n+1} = (\mathbf{v}_c^h)_n + \left((\mathbf{a}_c^h)_n + (\mathbf{a}_c^h)^i_{n+1} \right) \frac{\Delta t}{2}. \tag{27}$$

Here the subscript n is the time level, and the superscript i is the counter for the multiple corrections. We stop the corrections when

$$\frac{(\mathbf{v}_c^h)^{i+1}_{n+1} - (\mathbf{v}_c^h)^i_{n+1}}{(\mathbf{v}_c^h)^{i+1}_{n+1}} \leq 2 \times 10^{-2}. \tag{28}$$

At each time step, the PCT model requires the computation of the cloud mean position and size, and identification of the elements contained within the cloud volume. This is done with the search algorithm described in [49].

5.2 Parameters of the turbulent particle dispersion

For spherical clouds, in [3] one of the choices was

$$\sigma^2 = 2 \langle \|\mathbf{u}'\|^2 \rangle^h \tau_L^2 \left(1 - e^{-\frac{\tau_{SUPG}}{\tau_L}} \right) \left(\frac{t}{\tau_L} - \left(1 - e^{-\frac{t}{\tau_L}} \right) \right) + \sigma_0^2, \tag{29}$$

where $\tau_L = \max(\tau_{SUPG}, \tau_R)$. For ellipsoidal clouds, to define \mathbf{S} , we first define $\mathbf{u}^F = \mathbf{u}^h - \overline{\mathbf{u}^h}$, where the overbar indicates time-averaging after a developed solution is reached. Then we write the components of \mathbf{S} as

$$S_{ij}(\mathbf{x}, t) = 2 \sqrt{\langle (u_i^F)^2 \rangle \langle (u_j^F)^2 \rangle} \tau_L^2 \left(1 - e^{-\frac{\tau_G}{\tau_L}} \right) \cdot \left(\frac{t}{\tau_L} - \left(1 - e^{-\frac{t}{\tau_L}} \right) \right) + (S_{ij})_0, \tag{30}$$

where

$$\tau_L = \max(\tau_G, \tau_R), \tag{31}$$

$$\tau_G = C_\mu^{3/4} \sqrt{\frac{3}{2}} \max_i \left(\frac{(u_i^F)^2}{\varepsilon_{ii}^D} \right) \text{ (no sum in } \varepsilon_{ii}^D \text{)}, \tag{32}$$

with $C_\mu = 0.09$ and $\varepsilon_{ij}^D = 2\nu \frac{\partial u_i^F}{\partial x_j} \frac{\partial u_j^F}{\partial x_i}$ [72].

6 Erosion models and erosion thickness calculation

Some parts of this section, included for completeness, are from [3]. In Sect. 2 we described two scale-up approaches that drive the sequence of evolution steps. In the “threshold erosion value” approach, we specify e_{thr} , and we assume that when the scaled-up e_{max} reaches e_{thr} , the erosion is at a level to alter the aerodynamics from its current operation pattern. The nominal target geometry plays a role in determining e_{thr} . The e computed in a simulation associated with an evolution step depends on the current target geometry and the size and spatial distribution of the particles. We assume that all these remain unchanged during an evolution step, justifying the scale-up of e to the erosion distribution for that evolution step. In the “threshold operation period” approach, we specify an

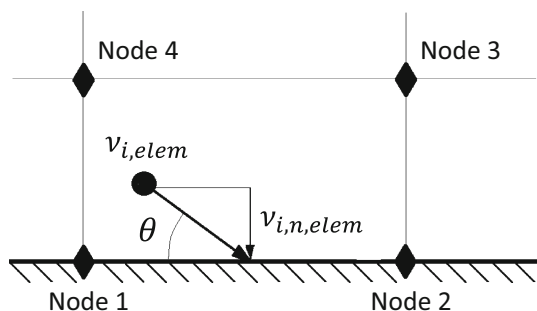


Fig. 1 Impact angle in the erosion model

operation period that we expect to be long enough to alter the aerodynamics, and that becomes the duration associated with each evolution step. We again assume that the current target geometry and the size and spatial distribution of the particles remain unchanged during an evolution step. The scale-up factor becomes the ratio of the number of particles in that duration to the number of particles used in the simulation. More details on the scale-up factors will be given in Sect. 6.3.

6.1 Erosion thickness computation

The erosion thickness, calculated at the element level, is expressed as

$$e = \frac{m_e}{\rho_t}, \tag{33}$$

where m_e is the eroded mass per unit area, computed in the simulation associated with the evolution step, and ρ_t is the density of the target material. Following the notation in [2,3], we obtain the eroded mass by summing up the mass eroded in each time step Δt :

$$m_e = \sum \Delta m_e, \tag{34}$$

where, after a threshold particle impact count is reached,

$$\Delta m_e = E \Delta n_p m_p. \tag{35}$$

Here E is the erosion rate, Δn_p is the particle impact count per unit area in Δt , and m_p is the particle mass. In the finite element PCT computation, Δn_p is calculated as

$$\Delta n_p = C_{elem} v_{i,n,elem} \Delta t, \tag{36}$$

where C_{elem} is the particle concentration in the element and $v_{i,n,elem}$ is the normal component of the particle impact velocity. Both are evaluated at the element center.

6.2 Erosion rate calculation

For E used in Eq. (35), we adopt the expression given in [73]:

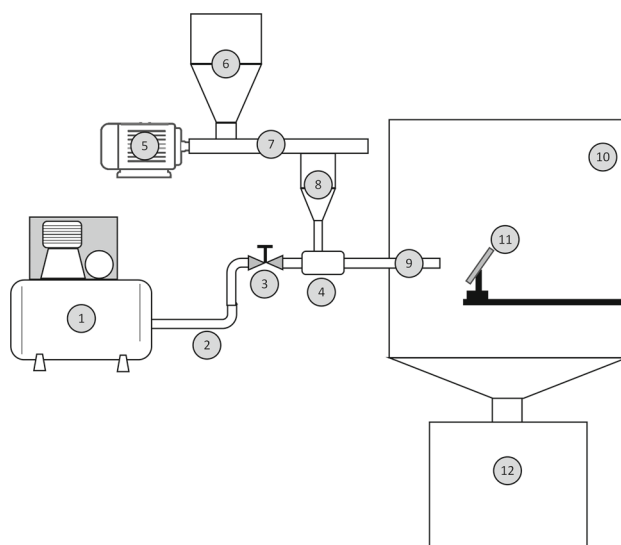


Fig. 2 Sketch of the erosion test bench: (1) reciprocating compressor; (2) air supply pipe; (3) pressure control valve; (4) mixing chamber; (5) electric motor; (6) particle tank; (7) cochlea; (8) particle inlet chamber; (9) injection pipe; (10) erosion chamber; (11) target; (12) particle collection chamber

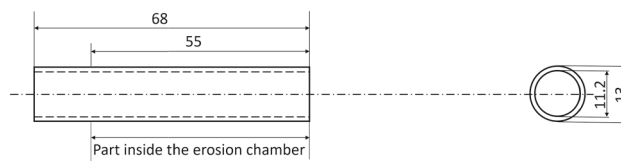


Fig. 3 Dimensions of the injection pipe (mm)

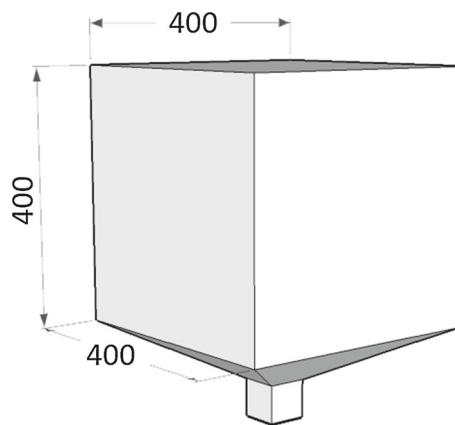


Fig. 4 Dimensions of the erosion chamber (mm)

$$E = K (v_{i,elem})^n, \tag{37}$$

where K and n are empirical coefficients that depend on the impact angle (see Fig. 1) and coating material. They are determined from the experiment (see Sect. 7). We note that this model has a double-dependency on the impact angle, in determining the value of K and n , and in calculating the impact count, through the parameter $v_{i,n,elem}$.

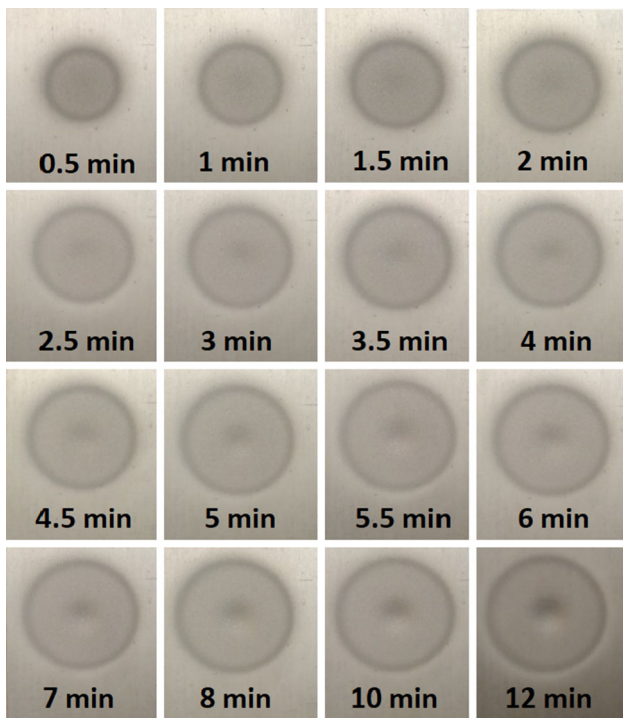


Fig. 5 Time evolution of the erosion zone

Table 1 Time evolution of the erosion rate

t_n (min)	M_n (g)	$(M_p)_n$ (g)	E_n (10^{-4} g/g)
0.0	60.2433	0.0	–
0.5	60.2219	115.8	1.848
1.0	60.2056	231.6	1.628
1.5	60.1891	347.4	1.560
2.0	60.1740	463.2	1.496
2.5	60.1586	579.0	1.463
3.0	60.1435	694.8	1.436
3.5	60.1284	810.6	1.417
4.0	60.1130	926.4	1.407
4.5	60.0972	1042.2	1.4018
5.0	60.0820	1158.0	1.3929
5.5	60.0666	1273.8	1.3872
6.0	60.0514	1389.6	1.3810
7.0	60.0216	1621.2	1.3675
8.0	59.9917	1852.8	1.3579
10.0	59.9314	2316.0	1.3467
12.0	59.8699	2779.2	1.3436

Here M : mass of the target, M_p : mass of the particles injected. The erosion rate is calculated based on the cumulative values $E_n = (M_0 - M_n)/(M_p)_n$

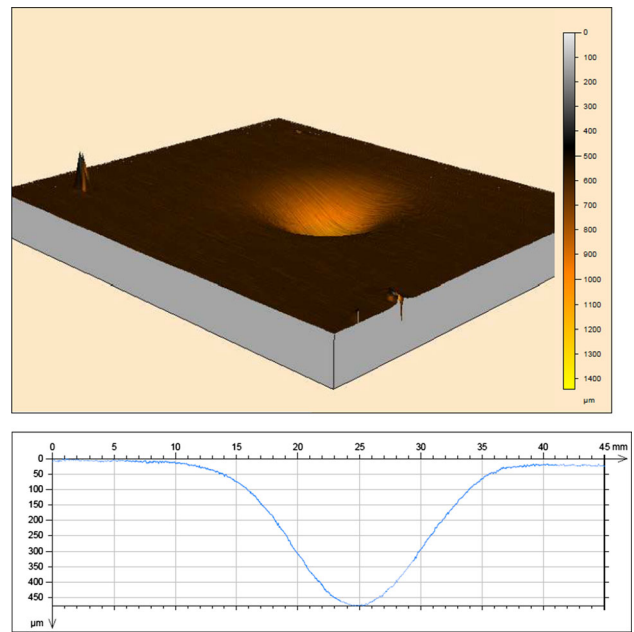


Fig. 6 Laser scan of the target at the end of the 40-min test. The lower picture shows the cross-section over the vertical plane passing through the injection-pipe axis

Table 2 Physical properties of the fluid and particle phases

Air density	1.225 kg/m ³
Particle density	4.0×10^3 kg/m ³
Particle diameter	1.9×10^{-4} m

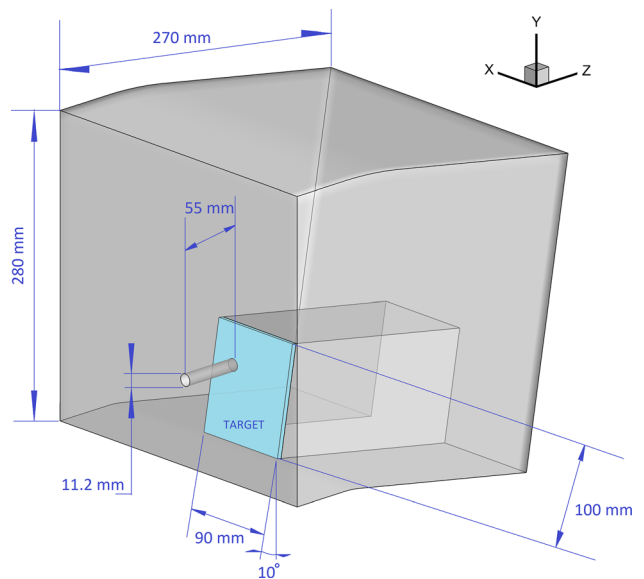


Fig. 7 Flow computation domain. The impingement location is 3 cm from the injection-pipe outlet and 0.8 cm off-center, closer to the right edge of the target

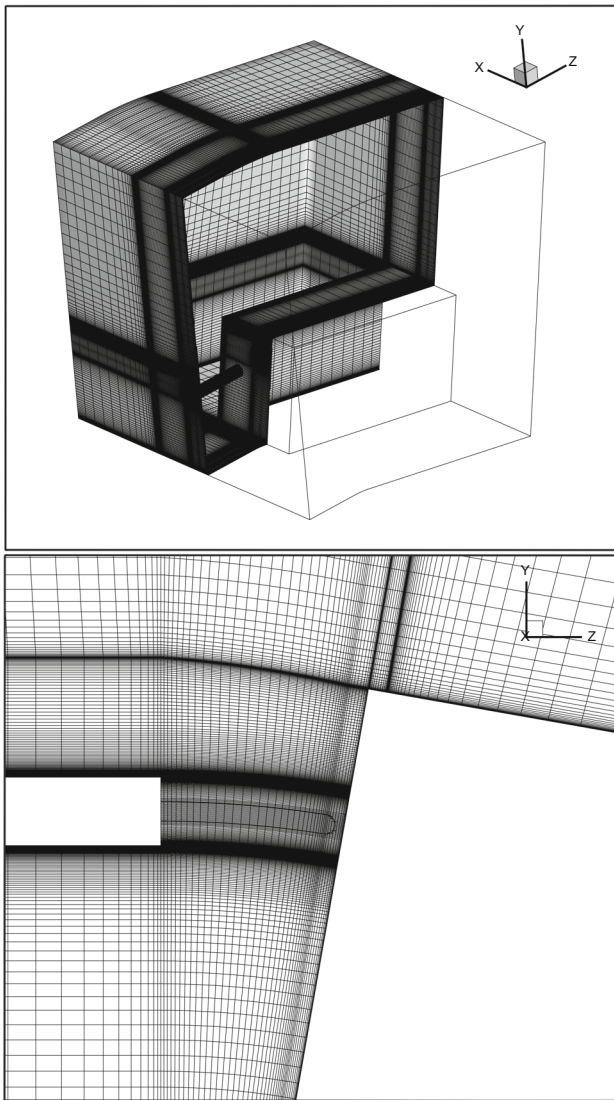


Fig. 8 Flow computation mesh

6.3 Erosion scale-up

In [3], it was assumed that the erosion is zero until a threshold impact count is reached. Here we use a simpler model by assuming that erosion occurs at any impact count. Then the scaled-up erosion thickness at an evolution step $i \geq 1$ becomes

$$e_{SU}^{(i)} = F_{ACT}^{(i)} e^{(i)} = F_{ACT}^{(i)} \frac{E^{(i)} m_p}{\rho_t} n_p^{(i)}, \tag{38}$$

with the scale-up factor $F_{ACT}^{(i)}$ determined from one of the two scale-up approaches described in Sect. 2. In the approach where we specify a threshold erosion value:

$$F_{ACT}^{(i)} = \frac{e_{thr}}{e_{max}^{(i)}}. \tag{39}$$

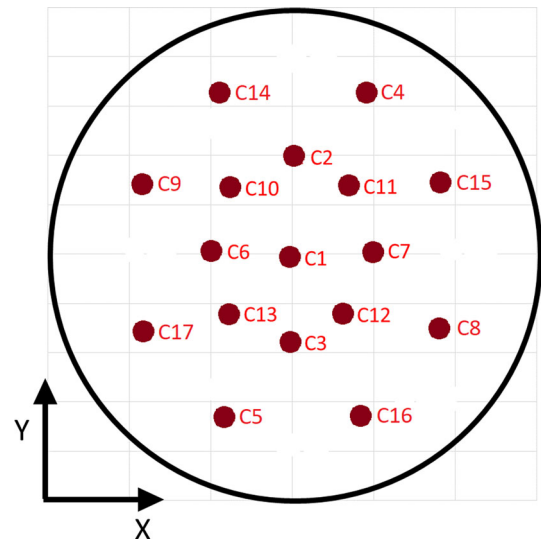


Fig. 9 Starting positions for the cloud centers

In the approach where we specify a threshold operation period:

$$F_{ACT}^{(i)} = \frac{n_R^{(i)}}{n_{TOT}}. \tag{40}$$

Here $n_R^{(i)}$ is the number of particles per unit area entering the PCT domain in the threshold operation period for the evolution step i , and n_{TOT} is the total number of particles per unit area entering the PCT simulation domain.

7 Experiments

7.1 Setup

The experimental setup is composed of four main sections: air supply system, particle supply system, mixing chamber, and erosion chamber (see Fig. 2). The air supply system is composed of a reciprocating compressor, an air supply pipe, and a pressure control valve that controls the air velocity. The particle supply system is composed of an electric motor, a particle tank, a cochlea, and a particle inlet chamber. The electric motor drives the cochlea that brings particles to the inlet chamber. The inlet chamber and pressure control valve bring particles and air, respectively, to the mixing chamber. Then the particles are injected through the injection pipe (see Fig. 3) in the erosion chamber (see Fig. 4). In this chamber, particles impact the target and are then collected to the particle collection chamber.

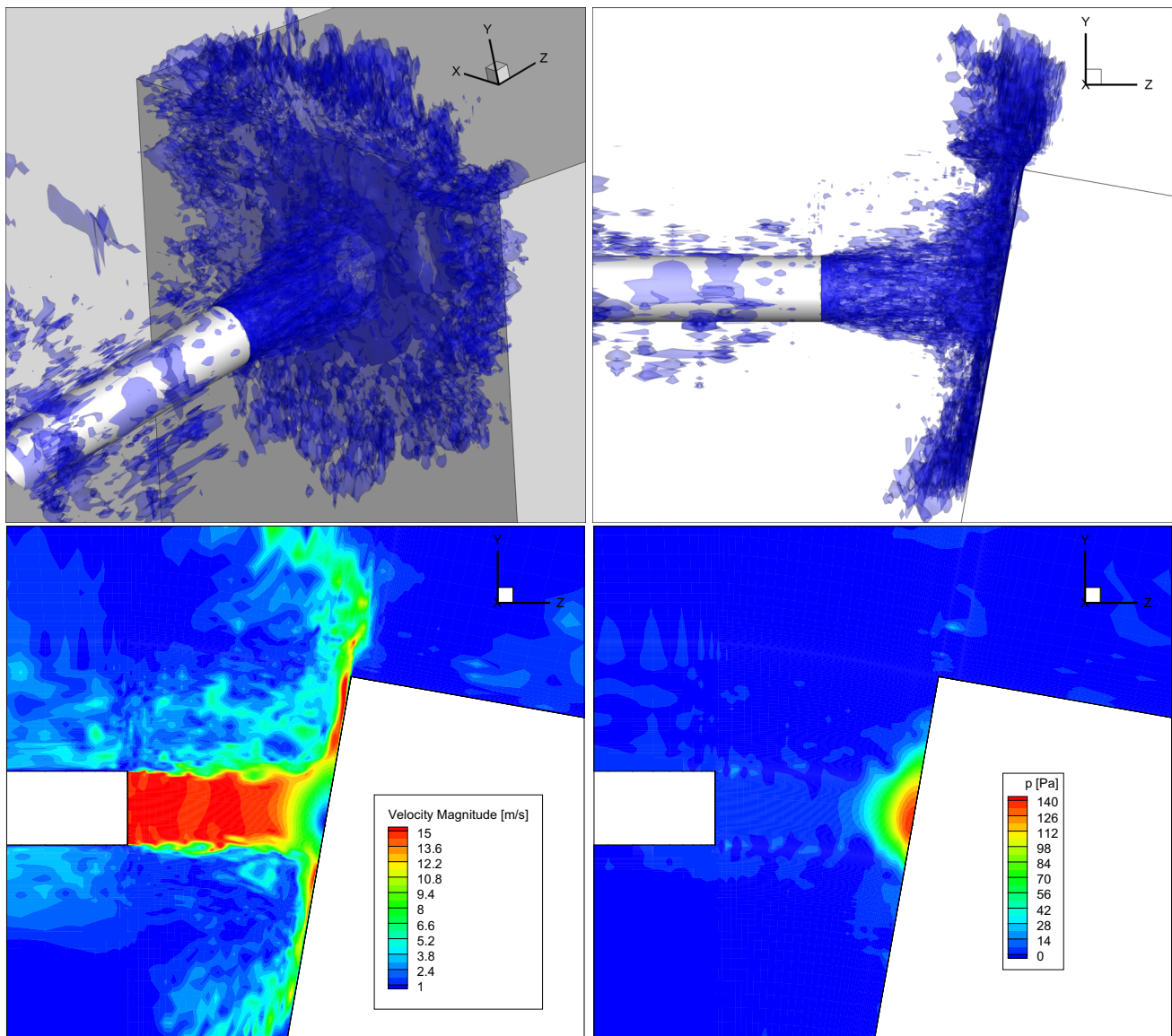


Fig. 10 Flow field in the first evolution step. Isosurfaces of $\|\mathbf{u}^h \times (\nabla \times \mathbf{u}^h)\|$ (top) and $\|\mathbf{u}^h\|$ (bottom-left) and p^h (bottom-right) on the $X = 0$ plane

7.2 Particles

Red-brown corundum particles are used in the experiment. They are essentially composed of Al_2O_3 and are quite suitable for erosion applications because of their ability to quickly abrade most materials. Moreover, because of their high toughness and impact resistance, they can be reused several times. Physical properties of the particles are given in “Appendix A” (Table 4).

7.3 Target material

The target is made of Peraluman EN AW 5754, an Al–Mg alloy. The physical properties are given in “Appendix A” (Table 5).

7.4 Test description

The injection velocity is 15 m/s and the impingement angle is 80° . The impingement location is 3 cm from the injection-pipe outlet and 0.8 cm off-center, closer to the right edge of the target. The test was carried on for 40 min. Figure 5 shows the time evolution of the erosion zone in the first 12 min of the test, and Table 1 shows the time evolution of the erosion

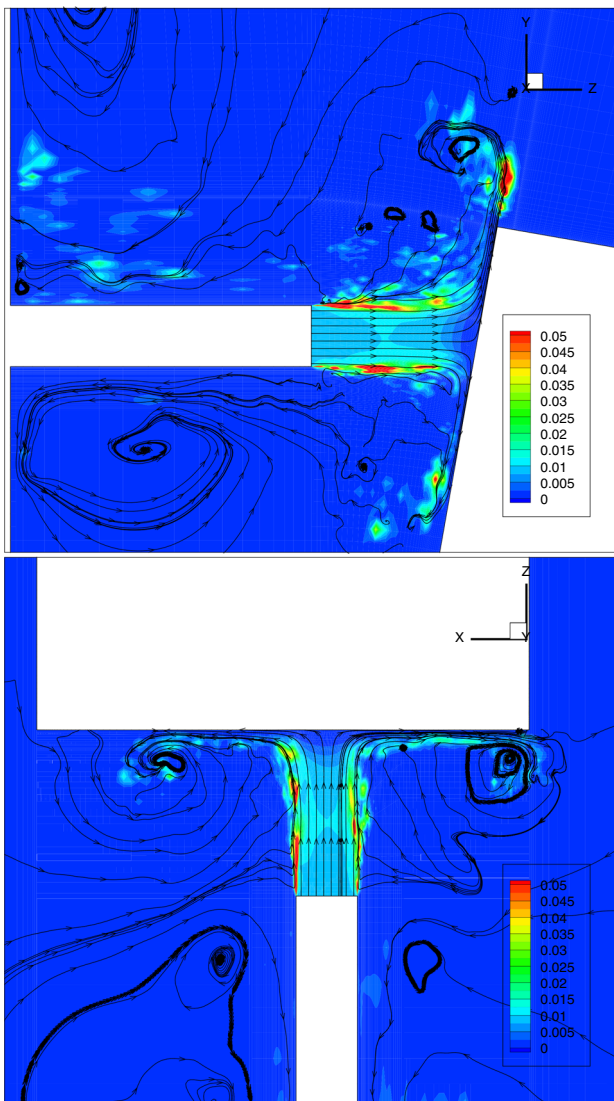


Fig. 11 Streamlines and $\overline{(u_3^F)^2}$ in the first evolution step, on the planes $X = 0$ (top) and $Y = 0$ (bottom)

rate in the same period. The eroded material was measured by weighing the target. Figure 6 shows the laser scan of the target at the end of the 40-min test.

We compare our computed data to the data from this test. In addition, we repeat the 40-min test for a set of impingement angles and injection velocities. We use the E values at the end of those tests to calculate, by curve fitting, K and n values (see Eq. (37)) for each impingement angle.

8 Computation

8.1 Problem setup

The computational setup reproduces the experimental setup. The injection velocity is 15 m/s and the impingement angle



Fig. 12 Streamlines and cloud trajectories in the first evolution step

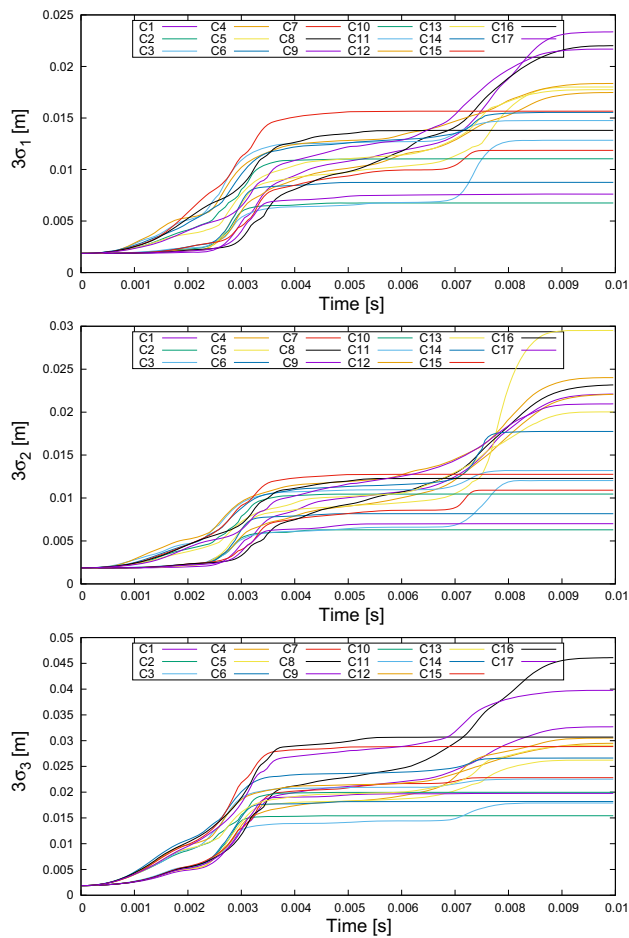


Fig. 13 Time-histories of the cloud ellipsoid semi-axis lengths ($3\sigma_1$, $3\sigma_2$, $3\sigma_3$) in the first evolution step. We note that the time scale is the PCT computation time scale

is 80° . The impingement location is 3 cm from the injection-pipe outlet and 0.8 cm off-center, closer to the right edge of the target. For the particle density and diameter we use the average values from the experiment. Table 2 shows the physical properties of the fluid and solid phases. The number of evolution steps is 5.

8.2 Computational domains, boundary conditions, and meshes

The flow computation domain is shown in Fig. 7. The origin of our coordinate frame is at the center of the target frontal surface. The boundary conditions are no-slip on the target and pipe outer surfaces, air jet injection velocity at the inflow boundary, zero-stress at the outflow boundary, and slip on all other boundaries. To reduce the computing time, we exclude the region downstream of the target.

The mesh, shown in Fig. 8, has 4 millions of hexahedral elements, with a $y^+ < 1$ for all wall elements in the jet

Table 3 Scale-up factors and spatially-maximum eroded mass per unit surface area ($(m_e)_{max}^{(i)}$) (kg/m^2)

i	$F_{ACT}^{(i)}$	$(m_e)_{max}^{(i)}$
1	73.947	3.462×10^{-3}
2	74.713	3.426×10^{-3}
3	75.133	3.407×10^{-3}
4	74.592	3.432×10^{-3}
5	74.314	3.445×10^{-3}

impingement area. The jet region has high refinement. The high-refinement zones we see in the interiors of the domain are mostly due to the block-structured nature of the mesh. We use the same mesh for the PCT computation.

We have 17 particle clouds, initially positioned at the injection-pipe outlet. Figure 9 shows the starting position for the cloud centers. The initial shape for all clouds is spherical with radius 1.87 mm. Each cloud has 1.0×10^5 particles.

8.3 Computational conditions

In the flow computations the time-step size is 1.0×10^{-6} s. The number of nonlinear iterations per time step is 5, and the number of GMRES iterations per nonlinear iteration is 30. In the PCT computations the time-step size is 1.0×10^{-5} s. In the first evolution step the flow computation is for 20,000 time steps. We extract the time-averaged quantities of Sect. 5 from the last 1000 steps. In the next 4 evolution steps the flow computation is for 10,000 time steps, and the time-averaged quantities are extracted again from the last 1000 steps. The PCT computation in all evolution steps is for 220 time steps.

8.4 Results

Figure 10 shows the flow field in the first evolution step. We see the turbulent nature of the impinging jet and a set of complex vortical structures. The jet enlarges immediately after the inlet, forming a conical volume with high turbulence production at the interface with the fluid at rest. Because the target is inclined, the flow is mostly upward after the impingement. We see a high-pressure region corresponding to the stagnation point, which is below the jet axis because the target is inclined. For the same reason, the elongation of the high-pressure region in the Y -direction is more in the upper part. Figure 11 shows the streamlines and $(u_3^F)^2$ in the first evolution step. As expected, the turbulence is higher at the interface between the jet and the fluid at rest.

Figure 12 shows the streamlines and cloud trajectories in the first evolution step. The trajectories are nearly symmetric with respect to the YZ plane. The cloud centers travel basically straight, carried by the air jet, and strike the target. After the rebound, the clouds on each side cross the YZ plane, while the clouds on the YZ plane follow a straight

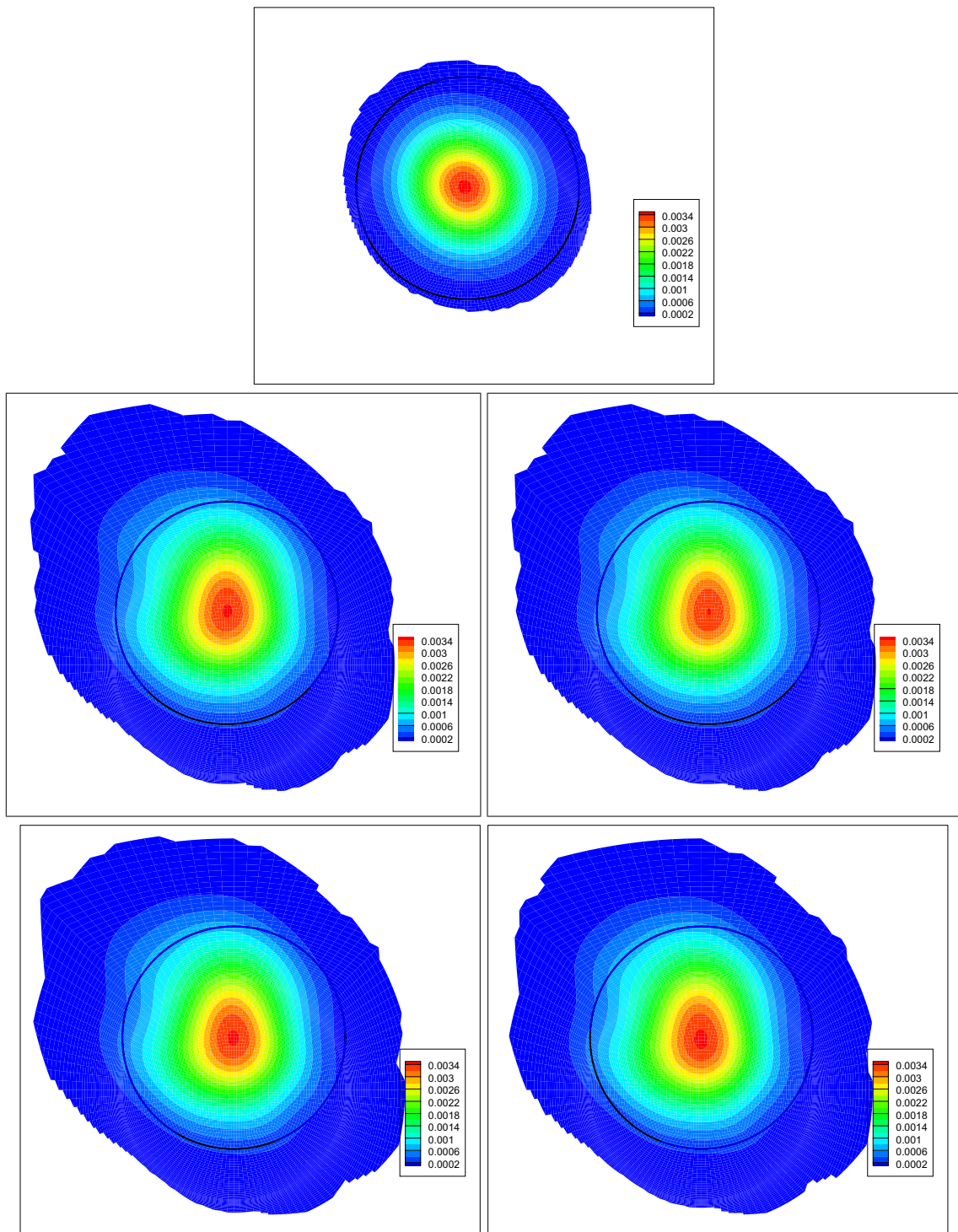


Fig. 14 Eroded mass per unit surface area (kg/m^2) at the end of each evolution-step PCT computation. The view is projected to the XY plane, and the black circle indicates the injection pipe

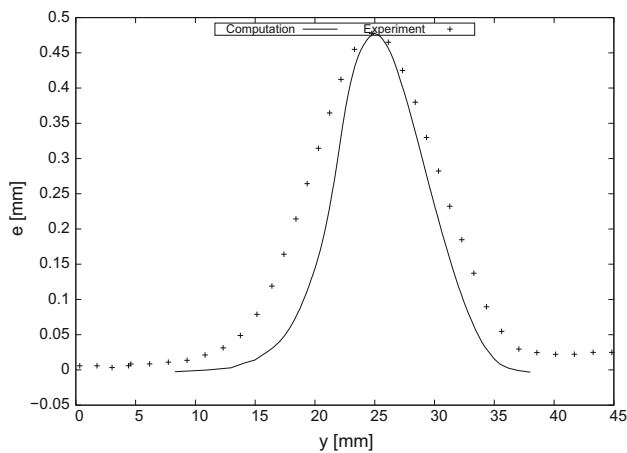


Fig. 15 Scaled-up erosion at the end of the last evolution step, cross-sectional view over the vertical plane passing through the injection-pipe axis

path. Figure 13 shows the time histories of the cloud ellipsoid semi-axis lengths in the first evolution step. We note that the time scale in the plots is the PCT computation time scale. We identify four phases.

- Phase 1: $t = 0$ s to $t \simeq 2 \times 10^{-3}$ s. All clouds are carried inside the jet streamtube. We see how the higher turbulence at the peripheries of the streamtube results in stretching of the clouds there, while the clouds near the tube center maintain their spherical shape until the impact.
- Phase 2: $t \simeq 2 \times 10^{-3}$ s to $t \simeq 3.3 \times 10^{-3}$ s. The particles impact the target. After the impact the particles enter a high-turbulence region of the flow field. That increases the particle dispersion, with a rapid increase in all three semi-axis lengths.
- Phase 3: $t \simeq 3.3 \times 10^{-3}$ s to $t \simeq 6.5 \times 10^{-3}$ s. The clouds, having exited the high-turbulence region, travel with little change in their shapes.
- Phase 4: $t \simeq 6.5 \times 10^{-3}$ s to $t \simeq 8.5 \times 10^{-3}$ s. Some of the clouds impact the back wall of the chamber. That increases the particle dispersion again.
- Phase 5: $t \simeq 8.5 \times 10^{-3}$ s to the end of the computation. The clouds, exiting again high-turbulence region, travel with little change in their shapes.

8.5 Erosion evolution

We set $e_{thr} = 95.6 \mu\text{m}$, which is 1/5 of the spatially-maximum erosion of $478 \mu\text{m}$ at the end of the test (see Fig. 6). Table 3 shows the scale-up factors and the spatially-maximum eroded mass per unit area $((m_e)_{max})$. Figure 14 shows the eroded mass per unit surface area at the end of each evolution-step PCT computation. Figure 15 shows the

scaled-up erosion at the end of the last evolution step together with the corresponding test data.

We can see from Fig. 14 that after an initial period of increase, the erosion area does not change much, and this is consistent with what we see in the experiment (see Fig. 5). Considering the complexity of the problem and the assumptions involved in the model, including the erosion-rate model, we think the discrepancy we see in Fig. 15 between the computed and experimental data is reasonable. Furthermore, we believe that some of the discrepancy is probably coming from not taking into account the dispersion caused by the particle–particle interactions.

9 Concluding remarks

We have presented an integrated method for computational analysis of particle-laden-airflow erosion. The analysis is valuable because erosion, e.g. on turbomachinery blades, over long periods of time, can degrade the aerodynamic performance. The analysis can help engineers have a better understanding of the erosion process, maintenance and protection of turbomachinery components. The analysis is challenging because it involves turbulent flows, large number of particles carried by the flow, turbulent dispersion of particles, and the target-geometry update due to the erosion has a very long time scale compared to the fluid–particle dynamics.

The main components of the integrated method are the VMS, a finite element PCT method, an erosion model based on two time scales, and the SEMMT. The turbulent-flow nature of the computational analysis is addressed with the VMS. The particle-cloud trajectories are calculated based on the time-averaged computed flow field and closure models defined for the turbulent dispersion of particles. One-way dependence is assumed between the flow and particle dynamics. Because the target-geometry update has a very long time scale, the update takes place in a sequence of “evolution steps” representing the impact of the erosion. A scale-up factor, calculated based on the update threshold criterion, relates the erosions and particle counts in the evolution steps to those in the PCT computation. The update threshold criterion is either a threshold erosion value that we expect to alter the aerodynamics of the target from its current operation pattern or a threshold operation period that we expect to be long enough to alter the aerodynamics of the target. As the target geometry evolves due to the erosion, the mesh is updated with the SEMMT, which not only protects the smaller elements from mesh deformation, but also protects the thin layers of elements near the target surface.

We presented a computation designed to match the sand-erosion experiment we conducted with an aluminum-alloy target. We showed that, despite the problem complexities

and model assumptions involved, we have a reasonably good agreement between the computed and experimental data. We also showed how updating the target geometry improves the quality of the computed data.

Acknowledgements The mathematical model and computational method parts of the work were supported in part by Grant-in-Aid for Challenging Exploratory Research 16K13779 from JSPS and Grant-in-Aid for Scientific Research (S) 26220002 from the Ministry of Education, Culture, Sports, Science and Technology of Japan (MEXT) (for the 5th author) and ARO Grant W911NF-17-1-0046 and Top Global University Project of Waseda University (for the last author).

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

Appendix A: Physical properties of the particles and target

See Tables 4 and 5.

Table 4 Physical properties of the red-brown corundum used in the experiments

Particle shape	Spidery
Hardness	9 Mohs, 2000 Knoop
Density	$3.9 \times 10^3 - 4.1 \times 10^3$ kg/m ³
Particle size	$1.8 \times 10^{-4} - 2.0 \times 10^{-4}$ m

Table 5 Physical properties of the target material Peraluman EN AW 5754

Yield strength (MPa)	130
Tensile strength (MPa)	220
Elongation (%)	8–11
Hardness (HB)	63
Density (kg/m ³)	2677.7

References

- Castorrini A, Corsini A, Rispoli F, Venturini P, Takizawa K, Tezduyar TE (2016) SUPG/PSPG computational analysis of rain erosion in wind-turbine blades. In: Bazilevs Y, Takizawa K (eds) Advances in computational fluid–structure interaction and flow simulation: new methods and challenging computations, Modeling and Simulation in Science, Engineering and Technology. Springer, pp 77–96. ISBN: 978-3-319-40825-5. https://doi.org/10.1007/978-3-319-40827-9_7
- Castorrini A, Corsini A, Rispoli F, Venturini P, Takizawa K, Tezduyar TE (2016) Computational analysis of wind-turbine blade rain erosion. *Comput Fluids* 141:175–183. <https://doi.org/10.1016/j.compfluid.2016.08.013>
- Castorrini A, Corsini A, Rispoli F, Venturini P, Takizawa K, Tezduyar TE (2019) Computational analysis of performance deterioration of a wind turbine blade strip subjected to environmental erosion. *Comput Mech* 64:1133–1153. <https://doi.org/10.1007/s00466-019-01697-0>
- Castorrini A, Venturini P, Corsini A, Rispoli F (2019) Numerical simulation of the blade aging process in an induced draft fan due to long time exposition to fly ash particles. *J Eng Gas Turbines Power* 141(1):011025
- Corsini A, Castorrini A, Morei E, Rispoli F, Sciuilli F, Venturini P (2015) Modeling of rain drop erosion in a multi-MW wind turbine. In: ASME Turbo Expo, Montreal, Canada
- Tezduyar TE, Takizawa K, Moorman C, Wright S, Christopher J (2010) Space-time finite element computation of complex fluid–structure interactions. *Int J Numer Methods Fluids* 64:1201–1218. <https://doi.org/10.1002/fld.2221>
- Rispoli F, Delibra G, Venturini P, Corsini A, Saavedra R, Tezduyar TE (2015) Particle tracking and particle-shock interaction in compressible-flow computations with the V-SGS stabilization and $YZ\beta$ shock-capturing. *Comput Mech* 55:1201–1209. <https://doi.org/10.1007/s00466-015-1160-3>
- Takizawa K, Tezduyar TE, Hattori H (2017) Computational analysis of flow-driven string dynamics in turbomachinery. *Comput Fluids* 142:109–117. <https://doi.org/10.1016/j.compfluid.2016.02.019>
- Komiya K, Kanai T, Otoguro Y, Kaneko M, Hirota K, Zhang Y, Takizawa K, Tezduyar TE, Nohmi M, Tsuneda T, Kawai M, Isono M (2019) Computational analysis of flow-driven string dynamics in a pump and residence time calculation. In: IOP conference series earth and environmental science, vol 240, p 062014. <https://doi.org/10.1088/1755-1315/240/6/062014>
- Kanai T, Takizawa K, Tezduyar TE, Komiya K, Kaneko M, Hirota K, Nohmi M, Tsuneda T, Kawai M, Isono M (2019) Methods for computation of flow-driven string dynamics in a pump and residence time. *Math Models Methods Appl Sci* 29:839–870. <https://doi.org/10.1142/S021820251941001X>
- Brooks AN, Hughes TJR (1982) Streamline upwind/Petrov–Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier–Stokes equations. *Comput Methods Appl Mech Eng* 32:199–259
- Tezduyar TE (1992) Stabilized finite element formulations for incompressible flow computations. *Adv Appl Mech* 28:1–44. [https://doi.org/10.1016/S0065-2156\(08\)70153-4](https://doi.org/10.1016/S0065-2156(08)70153-4)
- Tezduyar T (2001) Finite element interface-tracking and interface-capturing techniques for flows with moving boundaries and interfaces. In: Proceedings of the ASME symposium on fluid-physics and heat transfer for macro- and micro-scale gas-liquid and phase-change flows (CD-ROM), ASME Paper IMECE2001/HTD-24206. ASME, New York
- Tezduyar TE (2003) Stabilized finite element formulations and interface-tracking and interface-capturing techniques for incompressible flows. In: Hafez MM (ed) Numerical simulations of incompressible flows. World Scientific, London, pp 221–239. https://doi.org/10.1142/9789812796837_0013
- Stein K, Tezduyar TE, Benney R (2004) Automatic mesh update with the solid-extension mesh moving technique. *Comput Methods*

- Appl Mech Eng 193:2019–2032. <https://doi.org/10.1016/j.cma.2003.12.046>
16. Tezduyar TE, Sathe S, Keedy R, Stein K (2006) Space-time finite element techniques for computation of fluid-structure interactions. *Comput Methods Appl Mech Eng* 195:2002–2027. <https://doi.org/10.1016/j.cma.2004.09.014>
 17. Bazilevs Y, Takizawa K, Tezduyar TE (2013) *Computational fluid-structure interaction: methods and applications*. Wiley, New York
 18. Hughes TJR (1995) Multiscale phenomena: Green's functions, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles, and the origins of stabilized methods. *Comput Methods Appl Mech Eng* 127:387–401
 19. Hughes TJR, Oberai AA, Mazzei L (2001) Large eddy simulation of turbulent channel flows by the variational multiscale method. *Phys Fluids* 13:1784–1799
 20. Bazilevs Y, Calo VM, Cottrell JA, Hughes TJR, Reali A, Scovazzi G (2007) Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows. *Comput Methods Appl Mech Eng* 197:173–201
 21. Bazilevs Y, Akkerman I (2010) Large eddy simulation of turbulent Taylor-Couette flow using isogeometric analysis and the residual-based variational multiscale method. *J Comput Phys* 229:3402–3414
 22. Tezduyar TE, Osawa Y (2000) Finite element stabilization parameters computed from element matrices and vectors. *Comput Methods Appl Mech Eng* 190:411–430. [https://doi.org/10.1016/S0045-7825\(00\)00211-5](https://doi.org/10.1016/S0045-7825(00)00211-5)
 23. Tezduyar TE (2003) Computation of moving boundaries and interfaces and stabilization parameters. *Int J Numer Methods Fluids* 43:555–575. <https://doi.org/10.1002/flid.505>
 24. Tezduyar TE, Sathe S (2007) Modeling of fluid-structure interactions with the space-time finite elements: solution techniques. *Int J Numer Methods Fluids* 54:855–900. <https://doi.org/10.1002/flid.1430>
 25. Corsini A, Menichini C, Rispoli F, Santoriello A, Tezduyar TE (2009) A multiscale finite element formulation with discontinuity capturing for turbulence models with dominant reactionlike terms. *J Appl Mech* 76:021211. <https://doi.org/10.1115/1.3062967>
 26. Rispoli F, Saavedra R, Menichini F, Tezduyar TE (2009) Computation of inviscid supersonic flows around cylinders and spheres with the V-SGS stabilization and $YZ\beta$ shock-capturing. *J Appl Mech* 76:021209. <https://doi.org/10.1115/1.3057496>
 27. Corsini A, Iossa C, Rispoli F, Tezduyar TE (2010) A DRD finite element formulation for computing turbulent reacting flows in gas turbine combustors. *Comput Mech* 46:159–167. <https://doi.org/10.1007/s00466-009-0441-0>
 28. Hsu M-C, Bazilevs Y, Calo VM, Tezduyar TE, Hughes TJR (2010) Improving stability of stabilized and multiscale formulations in flow simulations at small time steps. *Comput Methods Appl Mech Eng* 199:828–840. <https://doi.org/10.1016/j.cma.2009.06.019>
 29. Corsini A, Rispoli F, Tezduyar TE (2011) Stabilized finite element computation of NOx emission in aero-engine combustors. *Int J Numer Methods Fluids* 65:254–270. <https://doi.org/10.1002/flid.2451>
 30. Takizawa K, Henicke B, Montes D, Tezduyar TE, Hsu M-C, Bazilevs Y (2011) Numerical-performance studies for the stabilized space-time computation of wind-turbine rotor aerodynamics. *Comput Mech* 48:647–657. <https://doi.org/10.1007/s00466-011-0614-5>
 31. Takizawa K, Henicke B, Puntel A, Spielman T, Tezduyar TE (2012) Space-time computational techniques for the aerodynamics of flapping wings. *J Appl Mech* 79:010903. <https://doi.org/10.1115/1.4005073>
 32. Corsini A, Rispoli F, Tezduyar TE (2012) Computer modeling of wave-energy air turbines with the SUPG/PSPG formulation and discontinuity-capturing technique. *J Appl Mech* 79:010910. <https://doi.org/10.1115/1.4005060>
 33. Corsini A, Rispoli F, Sheard AG, Tezduyar TE (2012) Computational analysis of noise reduction devices in axial fans with stabilized finite element formulations. *Comput Mech* 50:695–705. <https://doi.org/10.1007/s00466-012-0789-4>
 34. Takizawa K, Tezduyar TE, McIntyre S, Kostov N, Kolesar R, Habluetzel C (2014) Space-time VMS computation of wind-turbine rotor and tower aerodynamics. *Comput Mech* 53:1–15. <https://doi.org/10.1007/s00466-013-0888-x>
 35. Corsini A, Rispoli F, Sheard AG, Takizawa K, Tezduyar TE, Venturini P (2014) A variational multiscale method for particle-cloud tracking in turbomachinery flows. *Comput Mech* 54:1191–1202. <https://doi.org/10.1007/s00466-014-1050-0>
 36. Takizawa K, Tezduyar TE, Kuraishi T (2015) Multiscale ST methods for thermo-fluid analysis of a ground vehicle and its tires. *Math Models Methods Appl Sci* 25:2227–2255. <https://doi.org/10.1142/S0218202515400072>
 37. Takizawa K, Tezduyar TE, Mochizuki H, Hattori H, Mei S, Pan L, Montel K (2015) Space-time VMS method for flow computations with slip interfaces (ST-SI). *Math Models Methods Appl Sci* 25:2377–2406. <https://doi.org/10.1142/S0218202515400126>
 38. Cardillo L, Corsini A, Delibra G, Rispoli F, Tezduyar TE (2016) Flow analysis of a wave-energy air turbine with the SUPG/PSPG stabilization and discontinuity-capturing directional dissipation. *Comput Fluids* 141:184–190. <https://doi.org/10.1016/j.compfluid.2016.07.011>
 39. Takizawa K, Tezduyar TE, Otoguro Y (2018) Stabilization and discontinuity-capturing parameters for space-time flow computations with finite element and isogeometric discretizations. *Comput Mech* 62:1169–1186. <https://doi.org/10.1007/s00466-018-1557-x>
 40. Otoguro Y, Takizawa K, Tezduyar TE, Nagaoka K, Mei S (2019) Turbocharger turbine and exhaust manifold flow computation with the space-time variational multiscale method and isogeometric analysis. *Comput Fluids* 179:764–776. <https://doi.org/10.1016/j.compfluid.2018.05.019>
 41. Kuraishi T, Takizawa K, Tezduyar TE (2019) Tire aerodynamics with actual tire geometry, road contact and tire deformation. *Comput Mech* 63:1165–1185. <https://doi.org/10.1007/s00466-018-1642-1>
 42. Takizawa K, Ueda Y, Tezduyar TE (2019) A node-numbering-invariant directional length scale for simplex elements. *Math Models Methods Appl Sci* 29:2719–2753. <https://doi.org/10.1142/S0218202519500581>
 43. Castorriani A, Corsini A, Rispoli F, Takizawa K, Tezduyar TE (2019) A stabilized ALE method for computational fluid-structure interaction analysis of passive morphing in turbomachinery. *Math Models Methods Appl Sci* 29:967–994. <https://doi.org/10.1142/S0218202519410057>
 44. Baxter LL (1989) Turbulent transport of particles. Ph.D. thesis, Brigham Young University
 45. Baxter LL, Smith PJ (1993) Turbulent dispersion of particles: the STP model. *Energy Fuels* 7:852–859
 46. Venturini P (2010) Modelling of particle-wall deposition in two-phase gas-solid flows. Ph.D. thesis, Sapienza University of Rome, Italy
 47. Cardillo L, Corsini A, Delibra G, Rispoli F, Sheard AG, Venturini P (2015) Simulation of particle-laden flows in a large centrifugal fan for erosion prediction. In: 58th American society of mechanical engineers turbine and aeroengine congress, Düsseldorf, Germany
 48. Kaer SK (2001) Numerical investigation of ash deposition in straw-fired furnaces. Ph.D. thesis, Aalborg University, Denmark
 49. Corsini A, Marchegiani A, Rispoli F, Venturini P (1993) Predicting blade leading edge erosion in an axial induced draft fan. *ASME J Eng Gas Turbines Power* 115:435–444

50. Hussein MF, Tabakoff W (1974) Computation and plotting of solid particle flow in rotating cascades. *Comput Fluids* 2(1):1–15
51. Hamed AA, Tabakoff W, Rivir RB, Das K, Arora P (2005) Turbine blade surface deterioration by erosion. *J Turbomach* 127(3):445–452
52. Hamed A, Tabakoff W, Swar R, Shin D, Woggon N, Miller R (2013) Combined experimental and numerical simulations of thermal barrier coated turbine blades erosion. In: Technical report, NASA/TM-2013-217857, E-18648. NASA Glenn Research Center, Cleveland
53. Keegan MH, Nash D, Stack M (2013) On erosion issues associated with the leading edge of wind turbine blades. *J Phys D Appl Phys* 46(38):383001
54. Springer GS, Yang C-I, Larsen PS (1974) Analysis of rain erosion of coated materials. *J Compos Mater* 8(3):229–252
55. Oka YI, Okamura K, Yoshida T (2005) Practical estimation of erosion damage caused by solid particle impact: part 1: effects of impact parameters on a predictive equation. *Wear* 259(1):95–101
56. Castorini A, Corsini A, Morabito F, Rispoli F, Venturini P (2017) Numerical simulation with adaptive boundary method for predicting time evolution of erosion processes. In: ASME turbo expo 2017: turbomachinery technical conference and exposition. American Society of Mechanical Engineers, paper no. V02DT48A019
57. Tezduyar TE, Behr M, Mittal S, Johnson AA (1992) Computation of unsteady incompressible flows with the finite element methods: space-time formulations, iterative strategies and massively parallel implementations. In: New methods in transient analysis, PVP-Vol 246/AMD-Vol 143. ASME, New York, pp 7–24
58. Tezduyar T, Aliabadi S, Behr M, Johnson A, Mittal S (1993) Parallel finite-element computation of 3D flows. *Computer* 26(10):27–36. <https://doi.org/10.1109/2.237441>
59. Johnson AA, Tezduyar TE (1994) Mesh update strategies in parallel finite element computations of flow problems with moving boundaries and interfaces. *Comput Methods Appl Mech Eng* 119:73–94. [https://doi.org/10.1016/0045-7825\(94\)00077-8](https://doi.org/10.1016/0045-7825(94)00077-8)
60. Stein K, Tezduyar T, Benney R (2003) Mesh moving techniques for fluid–structure interactions with large displacements. *J Appl Mech* 70:58–63. <https://doi.org/10.1115/1.1530635>
61. Lain S, Sommerfeld M (2003) Turbulence modulation in dispersed two-phase flow laden with solids from a Lagrangian perspective. *Int J Heat Fluid Flow* 24:616–625
62. Wang LP (1990) On the dispersion of heavy particles by turbulent motion. Ph.D. thesis, Washington State University
63. Litchford LJ, Jeng SM (1991) Efficient statistical transport model for turbulent particle dispersion in sprays. *AIAA J* 29:1443–1451
64. Jain S (1995) Three-dimensional simulation of turbulent particle dispersion. Ph.D. thesis, University of Utah
65. Borello D, Venturini P, Rispoli F, Saavedra GZR (2013) Prediction of multiphase combustion and ash deposition within a biomass furnace. *Appl Energy* 101:413–422
66. Venturini P, Borello D, Iossa CV, Lentini D, Rispoli F (2010) Modelling of multiphase combustion and deposit formation and deposit formation in a biomass-fed boiler. *Energy* 35:3008–3021
67. Armenio V, Fiorotto V (2001) The importance of the forces acting on particles in turbulent flows. *Phys Fluids* 13:2437–2440
68. Schiller L, Naumann A (1933) Über die grundlegenden berechnungen bei der schwekrafstaubereitung. *Z Ver Deutsch Ing* 77:318–320
69. Tezduyar TE, Park YJ (1986) Discontinuity capturing finite element formulations for nonlinear convection–diffusion–reaction equations. *Comput Methods Appl Mech Eng* 59:307–325. [https://doi.org/10.1016/0045-7825\(86\)90003-4](https://doi.org/10.1016/0045-7825(86)90003-4)
70. Rispoli F, Corsini A, Tezduyar TE (2007) Finite element computation of turbulent flows with the discontinuity-capturing directional dissipation (DCDD). *Comput Fluids* 36:121–126. <https://doi.org/10.1016/j.compfluid.2005.07.004>
71. Otaguro Y, Takizawa K, Tezduyar TE (2020) Element length calculation in B-spline meshes for complex geometries. *Comput Mech* published online. <https://doi.org/10.1007/s00466-019-01809-w>, <https://doi.org/10.1007/s00466-019-01809-w>
72. Pope SB (2000) *Turbulent flows*. Cambridge University Press, Cambridge, UK
73. Arjula S, Harsha A, Ghosh M (2008) Solid-particle erosion behavior of high-performance thermoplastic polymers. *J Mater Sci* 43(6):1757–1768

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.